REMARKS

Claims 1–13 and 15–31 are currently pending in the application. In the above amendment, Applicants' representative has amended claims 1, 3-4, 7-7, 9-12, 20, 22, 24-26, 28, and 30-31 to more distinctly and clearly claim that which Applicants regard as their invention. In an Office Action dated June 25, 2004 ("Office Action"), the Examiner rejected claims 1 and 7 under 35 U.S.C. § 112, second paragraph, rejected claims 1–12 under 35 U.S.C. § 103(a) as being unpatentable over Sparks et al., U.S. Patent No. 5,212,784 ("Sparks") in view of Beier et al., U.S. Patent No. 6,065,018 ("Beier"), rejected claims 13 and 15–17 under 35 U.S.C. § 103(a) as being unpatentable over Carter et al., U.S. Patent No. 5,909,540 ("Carter") in view of Beier, and rejected claims 18–19 under 35 U.S.C. § 103(a) as being unpatentable over Carter in view of Beier and further in view of Mutalik et al., U.S. Patent No. 6,161,111 ("Mutalik"). Applicants' representative respectfully traverses these 35 U.S.C. § 112 and 35 U.S.C. § 103 rejections, for reasons provided below.

Applicants' representative wishes to address two anomalies in the Office Action. First, as noted above, the Examiner rejected claims 1–12, 13 and 15–17, and 18–19. However, the Examiner did not address claims 20–31 added by Applicants' representative in an Amendment filed on December 19, 2003. Applicants' representative requests that the Examiner please note the claims added in the Amendment filed December 19, 2003 and inform Applicants' representative of their current disposition in a subsequent Office Action. Second, on page 3 of the Office Action, the Examiner rejected claims 1–12 as being unpatentable over Sparks in view of Beier. However, on page 6, in specifically rejecting claims 11 and 12, the Examiner discusses a third reference, Sakuraba, which is neither listed in the Notice of References Cited nor in the Notice of References Cited in the previous Office Action dated October 1, 2003. Applicants' representative respectively requests clarification with regard to the specific rejection of claims 11 and 12 and, if possible, a patent number for the Sakuraba reference.

With respect to the Examiner's 35 U.S.C. § 112, second paragraph rejection, Applicants' representative has amended claims 1 and 7 and dependent claims depending from claims 1 and 7 to clarify that which Applicants regard as their invention. The Examiner states, in Section 3 of the Office Action, that:

> {I}t is unclear whether the Applicant refers to the logical device or mirror. It is unclear whether the current time stamp is the update time stamp in the second logical device. It is also unclear whether the current count is the updated count in the second logical device. The Examiner treats the mirror and the second logical device are the same, the current time stamp and the updated time stamp are the same, and the current count and updated count are the same for examination purposes.

Please consider amended claim 1, provided below, for the Examiner's convenience:

> 1. A method for backing up a computer-readable object stored on a first logical device unit, the method comprising:
>
> when the object is not currently mirrored to a mass storage device, creating a mirror for the object on a second logical device unit;
>
> when the object and the mirror for the object are split, resyncing the object with the mirror for the object;
>
> splitting the object and the mirror for the object so that the mirror becomes a backup copy of the object and so that I/O requests directed to the object are not automatically directed to the mirror;
>
> retrieving a first instance of a current timestamp from the second logical device unit and saving it as a saved timestamp;
>
> updating the current timestamp upon executing any I/O operation directed to the second logical device unit that alters data stored on the second logical device unit; and
>
> when the object is determined to need to be restored from the mirror,
>
> retrieving a second instance of the current timestamp from the second logical device unit;
>
> comparing the retrieved second instance of the current timestamp to the saved timestamp; and
>
> when the second instance of the current timestamp is equal to the saved timestamp, copying the mirror to the first logical device unit to replace or again create the object on the first logical device unit.

Applicants' representative can well understand why the Examiner may have been confused by the previous wording of claims 1 and 7. Applicants' representative endeavors, below, to summarize certain aspects of claims 1 and 7 that were not clear to the Examiner from the previous versions of claims 1 and 7. A logical device unit, abbreviated "LUN" in the current application, is not a mirror. A LUN is a virtual mass-storage device, as discussed in detail on the current application beginning on line 19 of page 3. A mirror, by contrast, is a second copy of a data object, or second data image of an object, stored within a computer system. A mirror may be stored on a LUN. However, a mirror is not equivalent to, or equal to, a LUN.

Mirrors are defined and discussed, in detail, in the current application beginning on line 15 of page 4.

Claim 1 claims one embodiment of the current invention described in the application. In the "retrieving a first instance of a current timestamp" step of claim 1, a current timestamp is obtained from the second logical device, or second LUN. As described concisely, beginning on line 21 of page 7, embodiments of the present invention employ LUN-level timestamping in which a timestamp is associated with each LUN and automatically updated by a disk-array controller that provides the LUN to remote computers. Next, in the "updating the current timestamp" step, the current timestamp is continuously updated when the second LUN is directed to alter any data stored on the second LUN. In other words, the current timestamp is continuously updated when the contents of the second LUN are changed or altered, for any reason. Again, update of the timestamp associated with a LUN by a disk-array controller is concisely and completely described in the current application, beginning on line 21 of page 7. Finally, when an object is determined to need to be restored from the mirror created on the second LUN, the system or application that restores the object again retrieves a second instance of the current timestamp from the second LUN, in the "retrieving a second instance of the current timestamp" step, compares, in the "comparing the retrieved second instance of the current timestamp" step, the retrieved second instance of the current timestamp to the timestamp saved by the application or system in a previous step and then, in the final step of claim 1, determines whether or not the object can be restored from the mirror by determining whether or not the second instance of the current timestamp retrieved from the second logical device is equal to the saved timestamp saved by the system or application in a previous step, indicating whether or not data within the second LUN has been altered, thereby corrupting the mirror. This process is concisely described, in the current application in the summary of the invention section, beginning on line 20 of page 7 and extending to line 8 of page 8. Saving of the timestamp returned by a LUN to a system or application is discussed, in detail, with respect to a C++-like pseudocode implementation included within the current application beginning on line 23 of page 22. Comparison of the saved timestamp with a current timestamp again retrieved from the LUN is described, in detail, beginning on line 13 of page 23.

To summarize, one embodiment of the present invention involves associating a timestamp with each LUN in a disk array by a disk-array controller. The

disk-array controller updates the timestamp associated with a LUN any time an operation is directed by the disk-array controller to the LUN that would result in altering data stored within the LUN. The disk-array controller returns a timestamp associated with a particular LUN to a remote system or application upon request. Therefore, a remote system or application can request the timestamp associated with a LUN at a point in time when the system or application intends to store a mirror copy on the LUN, and can later again request a timestamp associated with the LUN in order to determine whether or not there is a chance that the mirror stored on the LUN has been altered since the mirror was stored on the LUN. In this fashion, the system or application can easily determine whether or not the mirror has been potentially corrupted, and can therefore decide whether or not to use the mirror data to restore a data object.

Applicants' representative has read the references cited by the Examiner in the Examiner's 35 U.S.C. § 103(a) rejections, and cannot find any teaching, mention, or suggestion by any of the cited references alone, or in combination, of Applicants' claimed invention. Sparks teaches a system in which a CPU is coupled to a primary controller. The primary controller is, in turn, coupled by separate logical busses to several mass-storage devices and several backup controllers, in turn connected to several backup devices. This architecture is clearly shown in Figure 2 of Sparks. Sparks' described system uses logical bus protocols to direct data from the CPU either to the storage devices (4A and 4B in Figure 2) during normal operation, or to backup devices (6 in Figure 2) during a backup operation. Sparks does not appear to be related to disk arrays or disk-array controllers, does not once mention logical device units, and aside from providing a relatively primitive mirroring capability by redundantly storing data on the second storage device (4B in Figure 2) when data is directed to the first storage device (4A in Figure 2), appears to be essentially unrelated to Applicants' claimed invention. Mirroring is well known, and has been commercially available for at least 30 years.

Beier, by contrast, is directed to synchronization of recovery logs for recovering contents-related, but different databases (Beier, abstract). The error recovery logs are maintained separately by each database management system. As described beginning on line 37 of column 8 of Beier, a recovery log includes multiple entries, each entry including a timestamp. A database management system generally continuously adds a timestamp-containing entry to the recovery log each time the

database management system executes a database transaction. Beier's described method, beginning on line 64 of column 8 and continuing to line 18 of column 9, involves using recovery logs to update a primary databases for each of two different types of databases, but only recovery-log information for a relational database that has timestamps less than the timestamp associated with the last log entry for a hierarchical database is used for restoring the primary databases. Timestamps associated with recovery-log entries in database management systems has been well known and commercially available for at least 25 years. Note, however, that Beier does not discuss or mention LUNs, timestamps associated with LUNs, timestamps associated with any physical device, or anything else related to Applicant's claimed invention.

There is no teaching or suggestion for the combination of Sparks and Beier. Sparks neither mentions nor suggests database management systems, timestamps, or timestamp-based synchronization of multiple databases. Sparks does not mention a need for comparing the data stored on the two different storage devices in order to determine whether or not mirror data has been corrupted. Sparks discloses a bus-protocol-related method for automatically directing data to either storage devices or backup devices. Beier, by contrast, discusses employing timestamp values in recovery-log entries maintained separately by two different databases to determine how far to restore the two different databases in the event of a primary database failure.

Moreover, a combination of Beier's timestamp protocol with a disk-array controller would not produce a feasible data-storage system. In Beier, a separate timestamp is associated with each database transaction, and stored in a data file maintained by a database management system. In other words, in Beier, separate timestamps are determined and associated with each update event. Timestamps are not associated with physical devices. Again, error-recovery logs are well known in database management systems. By contrast, in Applicants' clearly claimed invention, a disk-array controller associates a single current timestamp with, and maintains the single current timestamp for, a LUN provided by the disk array-controller. A LUN is a virtual mass-storage device that, as described in the current application, may include only a portion of a single physical mass-storage device, or may span multiple physical mass-storage devices. The disk-array controller maintains a single current timestamp for a LUN and updates the timestamp each time the disk-array controller directs an operation to the LUN that may result in altering data stored within the LUN. Unlike

in Beier, where a timestamp is associated with each data-altering transaction and stored in a file, the disk-array controller may store the single timestamp associated with a LUN in memory. There is a good reason for the marked difference between the claimed use of timestamps in the current application and Beier's reference to well-known database recovery logs. If a disk-array controller were to timestamp and log each data-altering operation directed to a LUN, the amount of storage space needed to store the timestamp-associated entries in the file could likely vastly exceed the amount of space used to store the data. Furthermore, accessing and manipulating the information stored in such huge files would be impractical by any disk-array-efficiency standards. Database transactions are relatively high-level events, and may involve hundreds or thousands of separate mass-storage-device operations. Moreover, because of the need to maintain data consistency amongst multiple database-accessing users, a database management system has no choice but to log transaction events with timestamps, in order to later be able to disambiguate and serialize multiple concurrent use of the database by multiple users. By contrast, the operations directed by a disk-array controller to a LUN are low-level operations, proceed at enormous, overall rate-limiting speed, and are not subsequently analyzed to determine the sequential order of the operations' occurrence. In summary, associating a timestamp with each operation directed by a disk-array controller to a LUN would simply cripple the performance and capacity of a disk array. LUN-based timestamping by a disk array controller is completely unrelated to database-management-system error recovery logs.

Neither Sparks nor Beier teach, mention, or suggest using a timestamp to determine whether or not a mirror copy of a data object has been corrupted. There is no mention in Sparks of a need for determining whether or not the mirror data has been altered. Beier discusses database management systems, in which error recovery logs are employed either to roll back transactions of a database to some previous consistent state, or to roll forward a backup database in time to a consistent state. Database management systems do not use timestamps associated with physical or logical mass-storage devices to determine whether or not the fidelity of mirror copies has been maintained. Database management systems are not disk-array controllers, and are designed to be largely independent of physical data storage systems and configurations. There is no suggestion in either Sparks, Beier, or in the art, for combining Sparks and Beier in a manner suggested by the Examiner. Were

event-based logs of mass-storage-device operations to be created and maintained by a disk-array controller, the resulting disk array would be ridiculously slow, and would soon have almost no capacity for storing data.

Because the Examiner employs one or both of Beier and Sparks in the separate rejections of claims 13 and 15–19, Applicants' representative believes these rejections to also be unjustifiable, and that claims 13 and 15–19 are therefore also allowable. Applicants' representative has addressed the additional cited references, Carter and Mutalik, in previous responses. In Applicants' representative's opinion, these references are as equally unrelated to the currently claimed invention as Sparks and Beier.

All of the claims remaining in the application are now clearly allowable. Favorable consideration and a Notice of Allowance are earnestly solicited.

Respectfully submitted,
Robert Alan Cochran and Harald Burose
Olympic Patent Works PLLC

Robert W. Bergstrom
Registration No. 39,906

Enclosures:
Postcards (2)
Transmittal in duplicate

Olympic Patent Works PLLC
P.O. Box 4277
Seattle, WA 98194-0277
206.621.1933 telephone
206.621.5302 fax